FROM TYPOGRAPHIC DISTRACTION TO SELECTIVE TOKEN INTERVENTION IN CLIP

Anonymous authors
Paper under double-blind review

ABSTRACT

Typographic attacks can redirect CLIP's prediction toward a rendered word even when the depicted object remains visible. This creates a selective intervention problem: suppress the distracting cue while retaining the image evidence needed for recognition. We study where this interference enters the visual representation through paired clean and attacked images. Restoring clean patch representations at positions overlapping the rendered text largely reverses the shift toward the rendered label, while inserting attacked representations at the same positions induces it. These findings motivate Patch Token Suppression (PTS), which learns a typographic score for each patch and moves representations with high scores toward a centroid estimated from patches assigned low scores in the same image. On synthetic ImageNet attacks, PTS improves object recovery while largely preserving clean accuracy. Evaluations on attacks in real images and ten additional datasets show consistent reductions in text confusion. Ablations identify spatial localization as the principal contributor to robustness, with smaller effects from the replacement target and interpolation rule.

1 INTRODUCTION

Contrastive vision-language models such as CLIP learn a shared image–text embedding space and support zero-shot recognition across many visual tasks (Radford et al., 2021). The same alignment makes rendered words unusually influential: adding a word to an otherwise unchanged image can redirect the prediction from the depicted object toward the written concept (Goh et al., 2021; Materzyńska et al., 2022). A useful defense must suppress this local distraction without erasing the visual evidence needed to recognize the object.

Existing defenses modify global projections or model weights (Materzyńska et al., 2022; Ilharco et al., 2022), class prompts (Azuma & Matsui, 2023), input-specific global feature dimensions (Wang et al., 2024), or selected attention components (Hufe et al., 2026; Liu et al., 2026). These approaches reduce typographic sensitivity, but their intervention units do not directly specify which spatial tokens to modify in a given image. Typography changes position, scale, and style across inputs, while object evidence extends beyond the text region. This raises a more local question: can we identify the patch tokens carrying the distraction and intervene before their influence is consolidated in the global representation?

Changes between paired clean and attacked representations concentrate at patches overlapping the rendered text before reaching the class token. Restoring clean states there reverses much of the shift toward the written label; inserting attacked states induces it. Matched non-text controls have little effect.

Without a clean test counterpart, Patch Token Suppression (PTS) scores patches using a projector trained on paired images and fixed typographic and reference prototypes. High-scoring patches move toward a centroid of low-scoring patches from the same image; CLIP remains frozen.

The learned scores reach approximately 95% patch-localization AP on both backbones. Spatially permuting them weakens object recovery more than the tested replacement and interpolation changes. Across synthetic and real-image attacks, PTS improves object accuracy over frozen CLIP in every evaluated setting, with larger clean-accuracy losses on some fine-grained datasets.

We make the following contributions:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

- We localize typographic interference through activation patching in both directions, connecting changes at patches overlapping rendered text to the shift toward the rendered label with matched controls at positions without text.
- We introduce PTS, a defense that keeps the backbone frozen, learns where to intervene, and suppresses typographic evidence by moving selected patch representations toward a centroid estimated from the same image.
- We evaluate localization and robustness across two CLIP backbones, synthetic attacks, and benchmarks of attacks in real images, and isolate spatial correspondence as the main contributor among the tested intervention choices.

2 Related Work

2.1 Mechanistic Interpretability of CLIP Vision Encoders

CLIP aligns global image and text representations (Radford et al., 2021), and its visual encoder contains units responsive to depicted objects and written names (Goh et al., 2021). Gandelsman et al. attribute the image embedding to patches, layers, and attention heads (Gandelsman et al., 2024); Zaigrajew et al. extract semantic concepts at several levels with hierarchical sparse autoencoders (Zaigrajew et al., 2025). Both expose structure within CLIP's global embedding.

Dyslexify traces the emergence of typographic information with layerwise probes, ranks attention heads by typographic attention, and tests selected heads through ablation (Hufe et al., 2026). nTAS uses sampled concept directions and gradient attribution to identify heads with disproportionate lexical focus (Liu et al., 2026). Both derive attention circuits from auxiliary images. PTS instead localizes the patch positions to edit in each test image and estimates their replacement target from that image.

2.2 Robustness to Typographic Attacks

Other defenses alter the final image embedding with learned projections (Materzyńska et al., 2022), interpolate fine-tuned and original weights (Ilharco et al., 2022), or learn an extra class-prompt token while keeping CLIP frozen (Azuma & Matsui, 2023). MirrorCLIP uses an image and its horizontal reflection to construct dimension-wise masks for textual and visual factors without training (Wang et al., 2024). These global or prompt-level interventions provide comparisons for the patch-level mechanism studied here.

3 From Typographic Distraction to Local Intervention

The object often remains visible after a typographic attack, yet the prediction changes. We test whether this change is mediated by patch representations before reaching the final image embedding. We use OpenAI CLIP ViT-B/32 and a sample of 5,000 ImageNet-1K validation pairs balanced across classes, retaining the 3,076 pairs classified correctly when clean. Selection is independent of the attack outcome. Renderer metadata identifies text patches using glyph occupancy of at least 1%; strict control patches have zero occupancy. Figure 2a illustrates the paired input and selected positions. We compare corresponding clean and attacked residual stream vectors at the input embedding and after each visual Transformer block. We refer to these vectors as patch representations.

Observation: attack-induced drift concentrates at text-overlap patches. Figure 1 shows that the cosine distance between paired clean and attacked patch representations is larger at text-overlap patches than at strict non-text patches through the middle and late blocks. At the block 8 boundary, the mean distance is 0.827 at text-overlap patches, 0.051 at non-text patches, and 0.021 at the class token. Class-token distance remains comparatively small until the final block, where it reaches 0.271. A control with shuffled characters separates this pattern from recognition of the nominal label. With whitespace positions, font, size, draw position, color, and source image fixed, shuffling preserves the progression from local representations to the global output but reduces prediction of the target label to 0.40% across all 5,000 sampled pairs. Thus, localized representation drift also

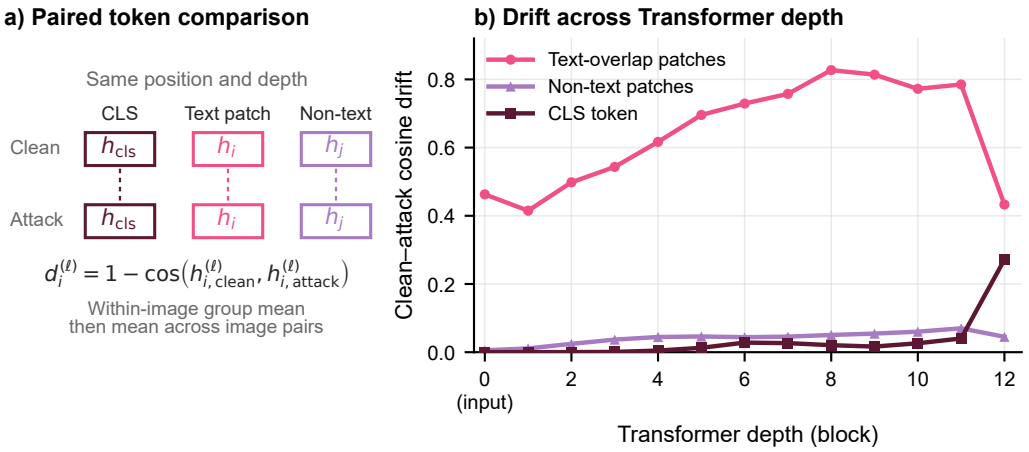


Figure 1: **Attack-induced representation drift concentrates at text-overlap patches.** (a) Corresponding clean and attacked tokens at each depth are compared by cosine distance. (b) Mean cosine distance for text-overlap patches (glyph occupancy at least 1%), strict non-text patches (zero occupancy), and the class token, averaged over the 3,076 clean-correct pairs. Depth 0 denotes input embeddings; depths 1–12 denote successive visual Transformer block outputs.

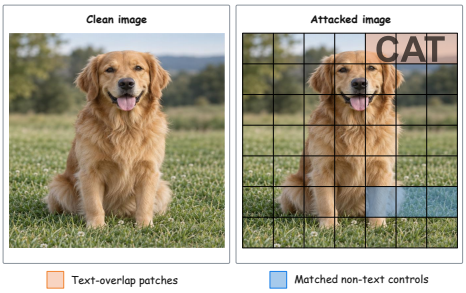
responds to typographic structure when the nominal word is disrupted. The next experiment tests whether the affected positions mediate the output change for attacks with intact words.

Intervention: activation patching at text positions changes the prediction. We perform activation patching in both directions, as summarized in Figure 2c. The target margin is the logit for the rendered label minus the logit for the true object, using cosine similarities scaled by 100. At each depth, we replace the patch representations at selected positions and run the remaining frozen blocks. The class token and all other patch positions remain unchanged. Each intervention is compared with a deterministic set of strict control positions of the same size, fixed across depths. We measure object classification accuracy (OCA) and the fraction of predictions assigned to the rendered label, or text confusion rate (TCR).

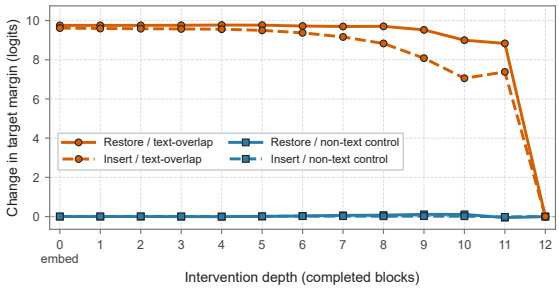
The solid curves in Figure 2b show the restoration intervention. Restoring patch representations at text positions substantially reduces the target margin, whereas restoring representations at matched control positions has little effect. At the block 6 boundary, restoration at text positions raises OCA from 57.44% to 94.57% and lowers TCR from 20.55% to 0.00%. The control obtains 57.83% OCA and 20.29% TCR. Replacing the patch representations at text positions with their clean counterparts therefore reverses most of the measured interference without restoring the rest of the image representation.

The dashed curves show the converse intervention: inserting attacked patch representations into the clean stream. At the block 6 boundary, insertion at text positions lowers OCA from 100.00% to 60.63% and raises TCR from 0.00% to 18.82%. The matched control insertion retains 98.54% OCA and 0.00% TCR, while complete attacked images obtain 57.44% OCA and 20.55% TCR. The two directions provide complementary evidence that these representations mediate the shift toward the rendered label in the paired synthetic setting. They transfer both character content and local rendering cues, so the identified unit is the patch representation at a position overlapping text. At depth 12, patch edits have no effect because no further attention block mixes them into the unchanged class token.

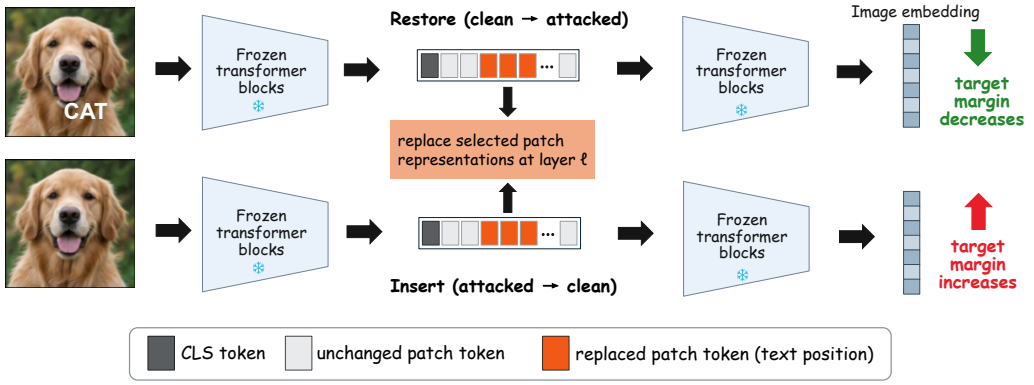
Design implication. The interventions in Figure 2 motivate editing patch representations at text-overlap positions. PTS learns to find those positions from paired data and constructs an image-specific centroid from low-scoring patches at inference.



(a) Input pair and patch selection.



(b) Intervention effects across depth.



(c) Bidirectional activation patching.

Figure 2: **Bidirectional activation patching localizes typographic interference.** (a) Renderer metadata identifies text-overlap patches in attacked images and matched non-text controls. (b) Solid curves show restoration and dashed curves insertion; orange uses text-overlap positions and blue matched controls. Effects are attack minus restored target margin and inserted minus clean target margin, respectively. (c) We restore clean source representations in the attacked stream or insert attacked source representations into the clean stream at the same positions and layer, leaving the class token and all other positions unchanged.

4 METHOD

Figure 3 shows the training and inference paths of PTS. A two-layer projector learns from paired clean and attacked patch representations grouped by the rendered text mask. After training, the projected embeddings yield fixed typographic and reference prototypes. At inference, prototype similarities score patches in the input image: low-scoring patches contribute more to the image-specific centroid, and high-scoring patches move farther toward it.

4.1 PROBLEM FORMULATION AND INTERVENTION POINT

Let x^{cln} be a clean image with visual class y, and let x^{atk} be a counterpart containing a rendered text distractor. For either image x, a frozen CLIP vision transformer produces one class token and $N = H_0W_0/P^2$ patch tokens at block ℓ, where P is the patch width. Writing the encoder prefix as $E_{1:\ell}$ and the normalized suffix as $F_{\ell+1:L}$, the intermediate representation and standard zero-shot prediction are

$$H^{(\ell)}(x) = E_{1:\ell}(x) = [h^{(\ell)}_{\mathrm{cls}}(x); h^{(\ell)}_1(x), \ldots, h^{(\ell)}_N(x)], \qquad \hat{y}(x) = \arg\max_{k \in \mathcal{C}} F_{\ell+1:L}(H^{(\ell)}(x))^\top t_k, \tag{1}$$

where t_k is the normalized text embedding for class k. Each $h^{(\ell)}_i(x)$ is the residual stream vector at patch position i, which we refer to as a patch representation. PTS replaces only these patch representations before applying the frozen suffix. The class token and all CLIP parameters remain unchanged. The objective is to recover y from x^{atk} while preserving zero-shot behavior on x^{cln}.

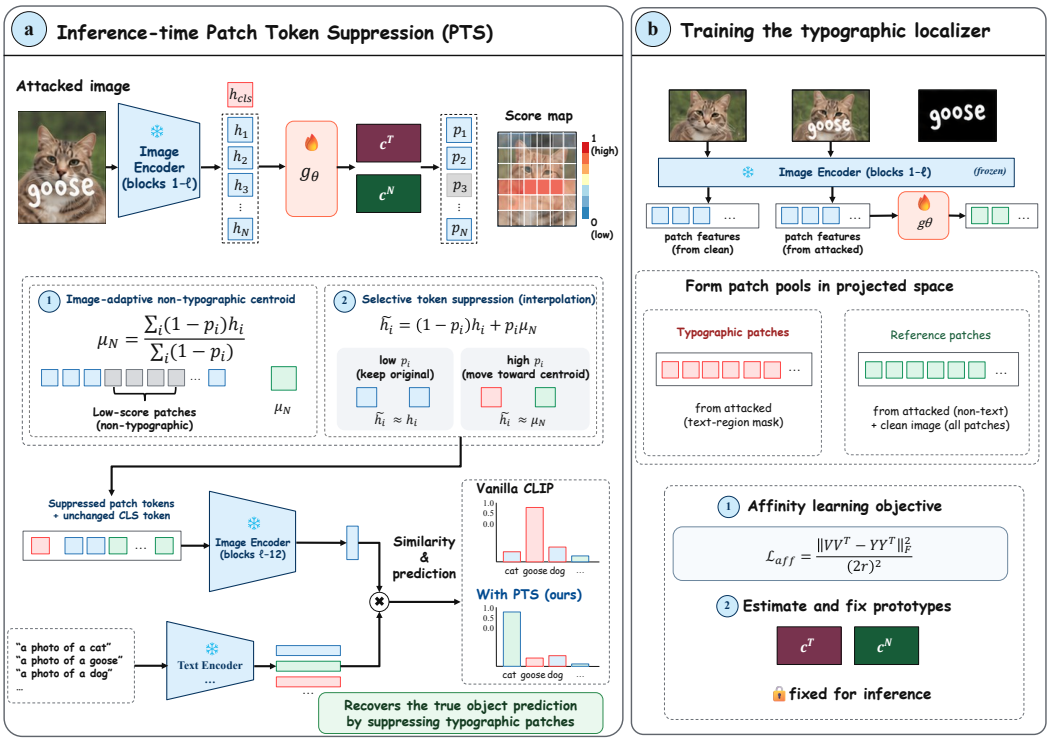


Figure 3: **Training and inference of Patch Token Suppression (PTS).** (a) At inference, the frozen CLIP vision encoder yields patch representations at block ℓ. The trained projector compares each patch with fixed typographic and reference prototypes to obtain a typographic score p_i. PTS uses weights $1 - p_i$ to estimate a non-typographic centroid from the same image, then interpolates each original patch representation toward that centroid with weight p_i. The class token is unchanged at this boundary; the remaining frozen blocks and CLIP class-text similarities produce the prediction. The bar charts are schematic. (b) During training, paired clean and attacked images and the rendered text-region mask define typographic patches (attacked text positions) and reference patches (attacked non-text and all clean positions). The affinity objective trains the projector; the resulting projected embeddings yield fixed typographic and reference prototypes for inference.

Each training tuple $(x_b^{\text{cln}}, x_b^{\text{atk}}, M_b)$ includes a renderer mask $M_b \in [0,1]^{H_0 \times W_0}$, reconstructed in model-input coordinates from the recorded rendering parameters. For the P^2-pixel region $\mathcal{R}_i$ associated with patch i, its text occupancy and binary patch label are

$$o_{bi} = \frac{1}{P^2} \sum_{u \in \mathcal{R}_i} M_b(u), \qquad a_{bi} = \mathbf{1}[o_{bi} \geq \rho] \in \{0,1\}, \tag{2}$$

where ρ is the occupancy threshold. If no patch reaches ρ, we label the maximum-occupancy patch as typographic. Neither y nor the rendered word identity supervises the projector. The mask provides only the location of rendered text.

4.2 Learning a Typographic Embedding

We learn a small projector $g_\theta : \mathbb{R}^d \to \mathbb{R}^{d_e}$ while keeping the CLIP backbone frozen:

$$v = g_\theta(h) = \text{Norm}(W_2\, \text{GELU}(W_1 h + b_1) + b_2)\,. \tag{3}$$

Here, $h \in \mathbb{R}^d$ is a patch representation at the intervention boundary, $v \in \mathbb{R}^{d_e}$, and $\text{Norm}(q) = q/\|q\|_2$. The normalized embedding v is used only for localization. Suppression is applied to the original residual stream representation h.

The paired images provide location supervision without requiring word recognition. For each pair, the typographic pool contains attacked patch representations with $a_{bi} = 1$; the reference pool contains unmasked attacked patch representations and all patch representations from the clean image.

Here, reference denotes the absence of the added distractor in the supervision. We sample r representations from each pool, using replacement when necessary, so that the larger reference region does not dominate training. For a mini-batch of B pairs, let $V_b \in \mathbb{R}^{2r\times d_e}$ contain the sampled embeddings and $Y_b \in \{0,1\}^{2r\times 2}$ their one-hot assignments. We optimize the deep clustering affinity objective (Hershey et al., 2016):

$$\mathcal{L}_{\text{aff}} = \frac{1}{B}\sum_{b=1}^{B} \frac{\|V_b V_b^\top - Y_b Y_b^\top\|_F^2}{(2r)^2}. \tag{4}$$

Because every row of V_b has unit norm, the target affinity is one for patches with the same assignment and zero otherwise. The loss therefore organizes the projected space by typographic presence without requiring the projector to predict the rendered word.

4.3 Localizing Typography with Prototypes

To turn the learned embedding into a score for an individual test patch, we represent each patch type by the normalized mean of its training embeddings (Snell et al., 2017). Define $v_{bi}^{\text{atk}} = g_\theta(h_i^{(\ell)}(x_b^{\text{atk}}))$ and $v_{bi}^{\text{cln}} = g_\theta(h_i^{(\ell)}(x_b^{\text{cln}}))$. Over the pairs used to estimate the prototypes, the fixed prototypes are

$$c_{\text{T}} = \text{Norm}\left(\sum_{b,i:a_{bi}=1} v_{bi}^{\text{atk}}\right), \quad c_{\text{N}} = \text{Norm}\left(\sum_{b,i:a_{bi}=0} v_{bi}^{\text{atk}} + \sum_{b,i} v_{bi}^{\text{cln}}\right), \tag{5}$$

where $c_{\text{T}}, c_{\text{N}} \in \mathbb{R}^{d_e}$ are normalized typographic and reference prototypes. For patch i in a test image x, let $v_i(x) = g_\theta(h_i^{(\ell)}(x))$. The softmax score over the two prototypes can be written as

$$p_i(x) = \sigma\left(\frac{v_i(x)^\top (c_{\text{T}} - c_{\text{N}})}{\tau}\right), \tag{6}$$

where σ is the logistic sigmoid and $\tau > 0$ is the softmax temperature. The condition $p_i(x) \geq 1/2$ holds exactly when the patch embedding is at least as similar to the typographic prototype as to the reference prototype. This provides a typographic score for each patch. It does not estimate whether a word is relevant to the recognition task.

4.4 Patch Token Suppression

The score $p_i(x)$ specifies where and how strongly to intervene. For compactness, write $h_i = h_i^{(\ell)}(x)$ and $p_i = p_i(x)$ in this subsection. To determine the replacement target, PTS estimates a weighted centroid from the patch representations of the same image. Let $w_i = 1 - p_i$ and $S_{\text{N}} = \sum_i w_i$. Then

$$\begin{aligned} \mu_{\text{N}}(x) &= \frac{\sum_{i=1}^{N} w_i h_i}{\max(S_{\text{N}}, \varepsilon)}, \\ \mu_{\text{N}}(x) &= \arg\min_{\mu\in\mathbb{R}^d} \sum_{i=1}^{N} (1-p_i)\|h_i - \mu\|_2^2 \quad \text{if } S_{\text{N}} > \varepsilon. \end{aligned} \tag{7}$$

For $S_{\text{N}} > \varepsilon$, the centroid is the weighted least squares representative of the original patch representations, with greater weight on patches assigned low typographic scores. It supplies a replacement target estimated from the visual evidence available at inference time. Section 5.3 compares this choice with zero replacement and a centroid estimated from the training set.

Finally, each patch representation is interpolated toward the centroid according to its typographic score:

$$\widetilde{h}_i = (1-p_i)h_i + p_i\mu_{\text{N}}(x) = h_i + p_i\big(\mu_{\text{N}}(x) - h_i\big). \tag{8}$$

A patch representation with a low score is nearly preserved, whereas one with a high score is largely replaced by the centroid. We keep $h_{\text{cls}}^{(\ell)}(x)$ unchanged and pass $[h_{\text{cls}}^{(\ell)}(x); \widetilde{h}_1, \ldots, \widetilde{h}_N]$ through the remaining frozen blocks. The spatial weights and replacement target are both determined from the test image.

Table 1: Patch localization on 50,000 synthetic ImageNet attacks (microaveraged, %). AP uses continuous scores; IoU and F1 use $p_i \geq 0.5$. Grid is the patch resolution.

Backbone	Grid	AP ↑	IoU ↑	F1 ↑
OpenAI ViT-B/32	7×7	94.45	76.18	86.48
OpenCLIP ViT-B/16	14×14	94.96	75.59	86.10

Implementation. Each backbone uses a projector with 0.1M trainable parameters, and the intervention follows the sixth transformer block. Appendix A reports the settings for optimization, thresholding, and prototype estimation.

5 Experiments

The experiments test three parts of the argument: whether the learned scores localize typography, whether their spatial correspondence drives robustness, and whether the resulting intervention improves recognition across backbones and attack datasets. This separates the quality of the localizer from the effect of the complete defense.

5.1 Experimental Setup

Models and training data. We evaluate OpenAI CLIP ViT-B/32 (Radford et al., 2021) and LAION-2B-pretrained OpenCLIP ViT-B/16 (Cherti et al., 2023). Each frozen backbone has a separate projector trained on 130,000 paired clean and synthetically attacked ImageNet-100 images.

Evaluation datasets. The common suite contains synthetic ImageNet validation attacks and three real-image attack sets: RTA-100, Disentangling, and PAINT. We also test transfer to ten clean datasets and their synthetic counterparts. ImageNet validation images are disjoint from projector training; Appendix B gives the splits and construction details.

Metrics. Following nTAS (Liu et al., 2026), we report *Object Classification Accuracy* (OCA ↑), top-1 accuracy for the depicted object under attack, and *Text Confusion Rate* (TCR ↓), the fraction classified as the rendered label. Clean performance uses top-1 accuracy; all values and absolute differences are in %. Benchmark evaluation matches label names, while the controlled mechanism and filter tests match class indices (Appendix A). Typographic entries use OCA/TCR.

Baselines. We compare with frozen CLIP, Defense-Prefix (Azuma & Matsui, 2023), MirrorCLIP (Wang et al., 2024), Dyslexify (Hufe et al., 2026), and nTAS (Liu et al., 2026) using shared dataset-specific class templates. Defense-Prefix uses its released prefix; Dyslexify and nTAS discover circuits separately for each backbone. Our nTAS† reproduction uses 1,280 ImageNet-100 training images rather than ImageNet-1K data for circuit discovery. Appendix A gives the configurations.

5.2 Evaluating Typographic Patch Localization

Using all 50,000 synthetic ImageNet validation attacks, disjoint from projector training, we test localization independently of classification. Renderer masks label patches at 1% glyph occupancy, as in training. Patch-level average precision (AP) uses continuous scores; IoU and F1 use $p_i \geq 0.5$. All metrics are microaveraged. Appendix A gives mask reconstruction details.

Localization across patch resolutions. Both separately trained projectors reach approximately 95% AP despite the different patch grids (Table 1). Their thresholded IoU is lower, near 76%, so the scores rank text-bearing patches well without perfectly isolating them. Appendix Figure 5 shows where elevated scores extend into the background. Localization is therefore a useful signal, but its effect on recognition depends on whether the high scores fall on the patches that carry the distraction.

Table 2: Controlled ViT-B/32 intervention ablations on 50,000 synthetic ImageNet attacks (%). Rows use class-index matching, unlike Table 3; the oracle uses renderer masks.

Variant	Scores	Replacement	Clean ↑	OCA ↑	TCR ↓
CLIP	—	—	62.03	39.58	21.97
Shuffled scores	shuffled	adaptive	61.62	42.30	12.86
Hard threshold	hard	adaptive	60.79	56.77	0.05
Zero replacement	soft	zero	60.72	55.32	0.04
Global centroid	soft	global	60.64	55.67	0.05
PTS	soft	adaptive	60.68	56.50	0.04
Renderer mask	oracle	adaptive	62.03	57.33	0.05

5.3 Ablating Patch Intervention

To test whether localization drives recognition, we vary score placement, thresholding, and replacement while holding the ViT-B/32 checkpoint, block-6 intervention, paired validation data, and evaluator fixed. We permute scores within each image, threshold them at 0.5, or use the renderer mask; separately, we replace the image-specific centroid with zero or a global centroid from 4,096 training pairs (Table 2). Permutation preserves the score distribution while breaking spatial correspondence.

Spatial correspondence drives recovery. Shuffling leaves each image's scores unchanged but assigns them to different patches. OCA drops from 56.50% to 42.30%, and TCR rises to 12.86% (Table 2). Hard thresholding and the tested replacement targets retain most of the recovery. The contrast isolates spatial correspondence as the dominant tested factor: recovery depends strongly on directing the intervention to the affected positions. Similar performance with hard scores also shows that soft interpolation is not essential in this setting.

The remaining clean-image gap. Exact renderer masks raise attacked OCA only slightly above PTS (57.33% vs. 56.50%) while preserving the frozen model's clean accuracy (Table 2). The learned scores thus capture most of the attacked-image benefit, but their less precise placement also changes some clean-image representations. This separates the benefit of selecting text-bearing patches from the remaining cost of imperfect localization.

5.4 Robustness Across Backbones and Datasets

Recovery on the common benchmarks. Across both backbones and all four attack sets, PTS improves object accuracy over frozen CLIP while keeping text confusion at or below 2.73% (Table 3). The ViT-B/16 Disentangling set illustrates that these are joint gains: OCA rises from 55.56% to 97.78% as predictions of the rendered label become rare. Clean ImageNet accuracy declines on both backbones, but by much less than the attacked-image gains. The intervention therefore recovers visual recognition without making the clean-accuracy cost disappear.

Cross-dataset transfer and clean accuracy. The same ViT-B/32 projector and fixed prototypes improve OCA on all ten transfer attack sets without dataset-specific retraining (Appendix Tables 6 and 7). The macro-average OCA gain is accompanied by a sharp drop in TCR, so transfer is visible in both recognition and resistance to the rendered label. Clean accuracy remains close to CLIP on eight datasets; the larger losses occur on FGVC Aircraft and Stanford Cars. Cars exposes a distinct failure mode: TCR falls to 0.32%, yet OCA reaches only 33.02%. Suppressing the distractor does not by itself restore the fine-grained distinctions needed to classify the car.

5.5 Comparison with Defenses against Typographic Attacks

Among the defenses with reported results, PTS has the lowest TCR in all eight backbone–attack comparisons and the highest or tied OCA in seven (Table 3). The exception is ViT-B/32 Disentangling: MirrorCLIP has slightly higher OCA, but its TCR is 12.22% against PTS's 2.22%. A defense can therefore reduce attraction to the written label without leading every object-accuracy comparison. Defense-Prefix and MirrorCLIP report no ViT-B/16 results in this table.

Table 3: Defenses on the common benchmarks (%). Clean is ImageNet accuracy; attacks report OCA/TCR (↑/↓) with label-name matching. Bold marks best values per backbone and metric. †: nTAS uses ImageNet-100 circuit-discovery data (Appendix A).

		Synthetic	Real-world typographic attacks		
Method	Clean	ImageNet	RTA-100	Disent.	PAINT
OpenAI CLIP ViT-B/32					
CLIP	62.15	39.66/22.03	46.70/34.40	45.56/46.67	50.91/31.82
Defense-Prefix	**62.63**	50.07/6.55	57.50/16.40	72.78/15.00	62.73/15.45
MirrorCLIP	62.44	53.10/1.00	**64.70**/3.10	**73.89**/12.22	66.36/5.45
Dyslexify	60.36	52.32/2.71	61.40/8.90	61.67/26.11	61.82/10.91
nTAS†	62.22	48.35/7.83	57.50/19.60	66.11/27.22	62.73/17.27
PTS	60.79	**56.62/0.04**	**64.70/0.10**	72.78/**2.22**	**70.00/0.00**
OpenCLIP ViT-B/16, LAION-2B					
CLIP	**69.39**	43.54/29.23	56.40/30.70	55.56/42.22	59.09/31.82
Dyslexify	68.38	60.94/2.80	69.10/11.70	88.89/8.89	75.45/10.00
nTAS†	68.76	58.96/5.86	66.70/15.00	90.00/10.00	70.00/19.09
PTS	68.60	**66.00/0.03**	**75.10/0.10**	**97.78/1.67**	**80.00/2.73**

The transfer comparison shows the same pattern across diverse image domains: PTS leads OCA on nine datasets and TCR on all ten (Appendix Table 7). EuroSAT is unusually vulnerable to rendered text, but removing it leaves PTS ahead of the next-best OCA baseline, Dyslexify. The average gain is therefore not driven by that single dataset. Stanford Cars is the OCA exception: Dyslexify reaches 34.40% against PTS's 33.02%. Together with the low TCR on Cars, this points to remaining object-recognition errors rather than continued attraction to the injected label.

6 Conclusion

Our activation patching experiments show that patch representations at text positions mediate typographic interference before the final class token readout. PTS uses this finding to learn where to intervene and draws replacement evidence from the input image itself. Across the evaluated backbones and attacks, it reduces text confusion and improves object recovery over frozen CLIP. The ablations locate the principal gain in spatial correspondence: changing which positions receive the scores matters far more than the tested replacement or interpolation choices. These results support spatially selective token intervention as a useful approach to typographic robustness.

Limitations and future work. Our evaluation focuses on contrastive CLIP encoders and image classification. Large vision-language models combine visual representations with language decoding, where rendered text may influence prediction through different multimodal pathways. Extending the analysis to visual question answering, captioning, and multimodal reasoning would clarify whether localized typographic interference and spatially selective intervention transfer beyond contrastive recognition.

AI use statement

We used generative AI tools to assist with manuscript drafting and readability edits, literature discovery, scientific figure preparation, and the interpretation and presentation of reported experimental results. We take responsibility for the final text, citations, numerical claims, and artifacts.

Reproducibility statement

Section 3 specifies the paired mechanism interventions and matched controls, Section 4 defines the learned intervention, and Section 5 reports the backbones, training data, evaluation sets, metrics, and baseline protocols. The appendix records optimization settings, mask construction, and filter-ablation controls. The accompanying code provides the core training and inference implementation of PTS, and all numerical tables were transcribed from saved aggregate metric files.

REFERENCES

Hiroki Azuma and Yusuke Matsui. Defense-prefix for preventing typographic attacks on CLIP. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 3644–3653, October 2023.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101: Mining discriminative components with random forests. In *European Conference on Computer Vision*, pp. 446–461, 2014. doi: 10.1007/978-3-319-10599-4_29.

Gary Bradski. The OpenCV library. *Dr. Dobb's Journal of Software Tools*, 25(11), November 2000.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2818–2829, 2023.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613, 2014. doi: 10.1109/CVPR.2014.461.

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, Yue Zhang, Wenyu Lv, Kui Huang, Yichao Zhang, Jing Zhang, Jun Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. PaddleOCR 3.0 technical report. *arXiv preprint arXiv:2507.05595*, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 178, 2004. doi: 10.1109/CVPR.2004.383.

Yossi Gandelsman, Alexei A. Efros, and Jacob Steinhardt. Interpreting CLIP's image representation via text-based decomposition. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5Ca9sSzuDp.

Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 6(3), 2021. doi: 10.23915/distill.00030. URL https://distill.pub/2021/multimodal-neurons/.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. doi: 10.1109/JSTARS.2019.2918242.

John R. Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 31–35, March 2016. doi: 10.1109/ICASSP.2016.7471631. URL https://www.merl.com/publications/TR2016-003.

Lorenz Hufe, Constantin Venhoff, Erblina Purelku, Maximilian Dreyer, Sebastian Lapuschkin, and Wojciech Samek. Dyslexify: A mechanistic defense against typographic attacks in CLIP. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=UI7mbsIZeN.

Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. In *Advances in Neural Information Processing Systems 35*, pp. 29262–29277, 2022. doi: 10.52202/068431-2122.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013. doi: 10.1109/ICCVW.2013.77.

Bohan Liu, Wenqian Ye, Guangzhi Xiong, Zhenghao He, Sanchit Sinha, and Aidong Zhang. Towards robustness against typographic attack with training-free concept localization, 2026. URL https://arxiv.org/abs/2607.02494.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Joanna Materzyńska, Antonio Torralba, and David Bau. Disentangling visual and written concepts in CLIP. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16410–16419, June 2022.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, pp. 722–729, 2008. doi: 10.1109/ICVGIP.2008.47.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012. doi: 10.1109/CVPR.2012.6248092.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30*, pp. 4077–4087, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/cb8da6767461f2812ae4290eac7cbc42-Abstract.html.

Tiancheng Wang, Yuguang Yang, Linlin Yang, Shaohui Lin, Juan Zhang, Guodong Guo, and Baochang Zhang. CLIP in mirror: Disentangling text from visual images through reflection. In *Advances in Neural Information Processing Systems 37*, pp. 24523–24546, 2024. doi: 10.52202/079017-0773.

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492, 2010. doi: 10.1109/CVPR.2010.5539970.

Vladimir Zaigrajew, Hubert Baniecki, and Przemyslaw Biecek. Interpreting CLIP with hierarchical sparse autoencoders. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 73918–73956. PMLR, 2025. URL https://proceedings.mlr.press/v267/zaigrajew25a.html.

A Additional Protocol Details

Training and inference settings. We use $\rho = 0.01$, $r = 8$, $\tau = 0.1$, and $\varepsilon = 10^{-6}$. Each projector is trained for 10 epochs with SGD, batch size 512, an initial learning rate of 2×10^{-3}, and cosine decay to 5×10^{-5}, without momentum or weight decay. Prototype estimation uses at most 4,096 paired training images. Both backbones use seed 300 and the final epoch-10 checkpoint; reported results are single-run estimates. The intervention follows block 6 (zero-based hook index 5), and the patch grid follows the backbone. The OpenCLIP model uses the `laion2b_s34b_b88k` pretrained weights.

Evaluation controls. The localization evaluation reconstructs glyph masks from the retained word, font, size, stroke, and placement metadata and uses the same 1% patch occupancy rule as training. The filter controls share the trained projector, prototypes, intervention block, prompts, and evaluation order. Only the score assignment or replacement specified in Table 2 changes. The global replacement is the mean of clean patch representations and attacked patch representations outside the rendered text from 4,096 training pairs, and shuffled scores use a deterministic permutation specific to each sample.

ImageNet scoring conventions. The benchmark evaluator compares predicted and target class names; the mechanism and filter evaluations compare class indices. Two names in the supplied ImageNet label list, *missile* and *sunglasses*, each correspond to two distinct classes. Name matching counts either same-named class as correct, whereas index matching distinguishes them. The filter evaluator also normalizes and scores all output features in float32; the benchmark evaluator retains native-precision scoring for CLIP and uses float32 for PTS. Tables 2 and 3 retain these separate evaluation conventions. Component effects are compared within Table 2, where all variants share class-index matching and float32 scoring.

Baseline reproduction. Defense-Prefix uses its released ViT-B/32 token and text encoder; MirrorCLIP follows the reference reflection-based decomposition. Dyslexify scores heads on Unsplash-10K and selects them using 130,000 ImageNet-100 training pairs, with clean-accuracy tolerance 0.01 and stopping after ten consecutive skips. nTAS† uses 1,280 ImageNet-100 training images, random-SAE expansion ratio 16, the final 20% of blocks, $\sigma = 1$, and $\alpha = 1$. Discovery is separate for each backbone and excludes evaluation images. Defense-Prefix and MirrorCLIP were not evaluated on ViT-B/16.

Candidate labels and prompts. The common suite uses 1,000 ImageNet, 100 RTA-100, 29 Disentangling, and 159 PAINT candidate labels. Real-world candidates are the union of object and rendered-text labels. Prompts use `a photo of a {class}.`, except Disentangling, which omits the second article. Defense-Prefix inserts its learned prefix before the class name. OCA and TCR use the same object-classification prediction. The broader benchmark retains dataset-specific prompts and candidates, including 101 labels for attacked ImageNet-100 (Table 6).

A.1 Intervention-Depth Sensitivity

We test every intervention depth in OpenAI CLIP ViT-B/32 by training a separate PTS projector at the post-residual output of each of the 12 visual Transformer blocks. Every run uses the same 130,000 paired ImageNet-100 training images, 10 training epochs, seed 300, and evaluation protocol. For each block, Figure 4 reports unweighted macro-averages over the ten clean datasets and their synthetic typographic counterparts in Table 7. OCA and TCR are computed from the same object-classification prediction.

Clean accuracy and OCA increase from blocks 1 to 6, from 56.92% to 62.68% and from 51.96% to 59.27%, respectively, while TCR decreases from 2.08% to 1.07%. Blocks 6–10 form a stable region: OCA remains between 59.03% and 60.09%, and TCR remains between 1.07% and 1.16%. The final block behaves differently: clean accuracy remains high at 63.62%, but OCA falls to 36.14% and TCR rises to 34.36%. The block-6 setting lies at the start of this plateau and attains the lowest macro-average TCR. At block 12, the edited patches have no remaining attention layer through which to affect the class token, which PTS leaves unchanged at the intervention boundary. The useful intervention depths therefore precede the final aggregation into the image embedding, consistent with the activation-patching results in Figure 2b.

B Datasets

We use the datasets below for projector training, the common typographic attack benchmark, and transfer across datasets. Counts refer to the evaluated splits rather than the complete source datasets.

ImageNet and ImageNet-100. ImageNet contains 1,000 object categories (Deng et al., 2009). We train each projector on 130,000 paired clean and synthetically attacked images drawn from 100

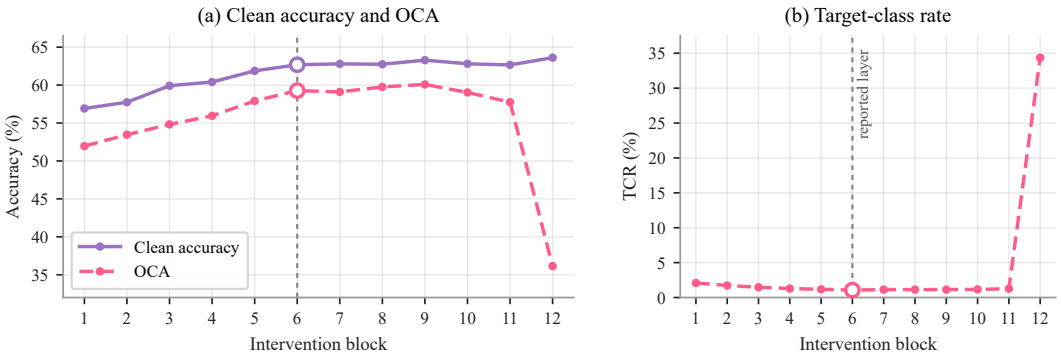


Figure 4: ViT-B/32 intervention-depth sweep. A separate PTS projector is trained after each visual Transformer block. Curves report unweighted macro-averages over ten datasets. The dashed vertical line marks block 6, used for the reported configuration. Left: clean top-1 accuracy and object classification accuracy under attack (OCA; higher is better). Right: target-class rate under attack (TCR; lower is better).

ImageNet classes. The common benchmark uses all 50,000 ImageNet validation images and one rendered attack for each image. For transfer evaluation, we use a balanced ImageNet-100 validation split with 5,000 images, or 50 per class, and the corresponding ImageNet-100-Text attack set, which also contains 5,000 images (Liu et al., 2026).

RTA-100. RTA-100 contains 1,000 real-world typographic attack images across 100 object categories, collected by attaching tags with incorrect class names to objects and photographing them (Azuma & Matsui, 2023).

Disentangling. The Disentangling dataset contains 180 images with written words placed over depicted objects and was introduced to probe the separation of visual and written concepts in CLIP (Materzyńska et al., 2022).

PAINT. PAINT contains 110 images with typographic attacks constructed from written notes (Ilharco et al., 2022).

For the following nine datasets, we generate one typographic counterpart for every clean image using the released MirrorCLIP renderer with seed 300 (Wang et al., 2024). Each pair retains the image and object label while the attacked version adds a competing class name.

Caltech-101. Caltech-101 covers 101 diverse object categories (Fei-Fei et al., 2004). We exclude the background category and evaluate 8,677 images.

Food-101. Food-101 contains 101 food categories with 1,000 images per category (Bossard et al., 2014). We evaluate its official test split, which contains 25,250 images.

Flowers-102. Flowers-102 contains 102 flower categories for detailed category recognition (Nilsback & Zisserman, 2008). We evaluate its test split of 6,149 images.

Oxford Pets. Oxford-IIIT Pets contains images of 37 cat and dog breeds (Parkhi et al., 2012). We evaluate its test split of 3,669 images.

EuroSAT. EuroSAT contains Sentinel-2 satellite images from ten land use and land cover categories (Helber et al., 2019). We evaluate the test split of 8,100 images used by the reproduced benchmark.

DTD. The Describable Textures Dataset contains 47 texture categories collected from images in the wild (Cimpoi et al., 2014). We evaluate 1,880 test images.

Stanford Cars. Stanford Cars contains 196 car categories defined by make, model, and year (Krause et al., 2013). We evaluate its test split of 8,041 images.

FGVC Aircraft. FGVC Aircraft contains 100 aircraft variants (Maji et al., 2013). We evaluate its test split of 3,333 images.

SUN397. SUN397 spans 397 indoor and outdoor scene categories (Xiao et al., 2010). We evaluate its test split of 19,850 images.

C PaddleOCR Region Blurring Baseline

Following the image preprocessing baseline introduced with Dyslexify (Hufe et al., 2026), we compare PTS with OCR region blurring on the ImageNet-100 class subset used by that work. This class subset is distinct from the nTAS ImageNet-100 transfer set in Tables 6 and 7. The evaluation contains 5,000 validation images, with 50 images from each class, and one typographic counterpart per image. We render a randomly selected competing class name after resizing each image to 224×224 pixels. The renderer uses Times New Roman, Georgia, or Arial, an initial font size of 30 pixels that is reduced until the word fits, and a random color and position. We use seed 300 for the complete set.

For each clean or attacked image, the PP-OCRv5 server detector from PaddleOCR returns polygonal text regions (Cui et al., 2025). We convert each polygon to its axis aligned pixel bounds and apply an OpenCV Gaussian blur only within that region, using a 51×51 kernel and $\sigma = 25$ (Bradski, 2000). The detector finds at least one region in 918 of the 5,000 clean images and 4,983 of the 5,000 attacked images. We evaluate OpenAI CLIP ViT-B/32 and OpenCLIP ViT-B/16 with LAION-2B weights. Each table reports top-1 object classification accuracy over 100 candidate labels.

Table 4: ImageNet-100 clean accuracy (%). Bold and underlined values mark the best and second best result in each column.

Setting	OpenAI ViT-B/32	OpenCLIP ViT-B/16
Normal	**72.26**	**74.74**
OCR blur	<u>71.76</u>	73.86
PTS	71.50	<u>74.50</u>
Both	71.08	73.44

Table 5: ImageNet-100 typographic attack accuracy (%). Bold and underlined values mark the best and second best result in each column.

Setting	OpenAI ViT-B/32	OpenCLIP ViT-B/16
Normal	42.72	46.56
OCR blur	65.26	68.52
PTS	**66.36**	**72.10**
Both	<u>65.64</u>	<u>68.90</u>

Table 5 shows that OCR region blurring raises attacked accuracy from 42.72% to 65.26% for ViT-B/32 and from 46.56% to 68.52% for ViT-B/16. PTS reaches 66.36% and 72.10%, respectively. These gains compare interventions at two stages: OCR blurring modifies detected pixel regions before encoding, whereas PTS modifies intermediate patch representations. Combining them yields 65.64% and 68.90%, below the 66.36% and 72.10% obtained with PTS alone. Thus, applying OCR blurring before PTS provides no additional recovery over the internal intervention on this benchmark.

The clean results in Table 4 distinguish the two backbones. For ViT-B/32, OCR blurring retains slightly more clean accuracy than PTS (71.76% vs. 71.50%). For ViT-B/16, PTS achieves both higher attacked accuracy and higher clean accuracy than blurring, with clean accuracy of 74.50%

vs. 73.86%. The combined condition has the lowest clean accuracy on both backbones, at 71.08% and 73.44%. Applying both edits therefore reduces clean recognition as well as attack recovery relative to PTS alone.

D Additional Localization Examples

Figure 5 compares prototype score maps on the same four attacked images, sampled across the per-image ViT-B/32 IoU distribution on 50,000 validation images. The maps show the continuous scores used for suppression; row labels report thresholded localization quality.

Spatial alignment. Both backbones place their strongest responses on the rendered words as the text moves from the upper part of the image to the bottom. In the first row, ViT-B/32 recovers all eight patches spanning two lines of text with no extra detections. In the third row, ViT-B/16 follows the narrow horizontal word more closely, reaching an IoU of 0.82 compared with 0.71 for ViT-B/32. Localization operates at patch resolution, so each marked token also represents visual content surrounding the glyph.

Errors outside the rendered mask. The lowest-IoU row detects all three marked ViT-B/32 patches but also selects three unmarked patches, yielding an IoU of 0.50. ViT-B/16 detects all four of its marked patches and selects five extra patches. These extra responses affect both operations in PTS: an unmarked patch with a high score contributes less to the centroid and moves more strongly toward it. Localization precision therefore affects preservation of the image representation.

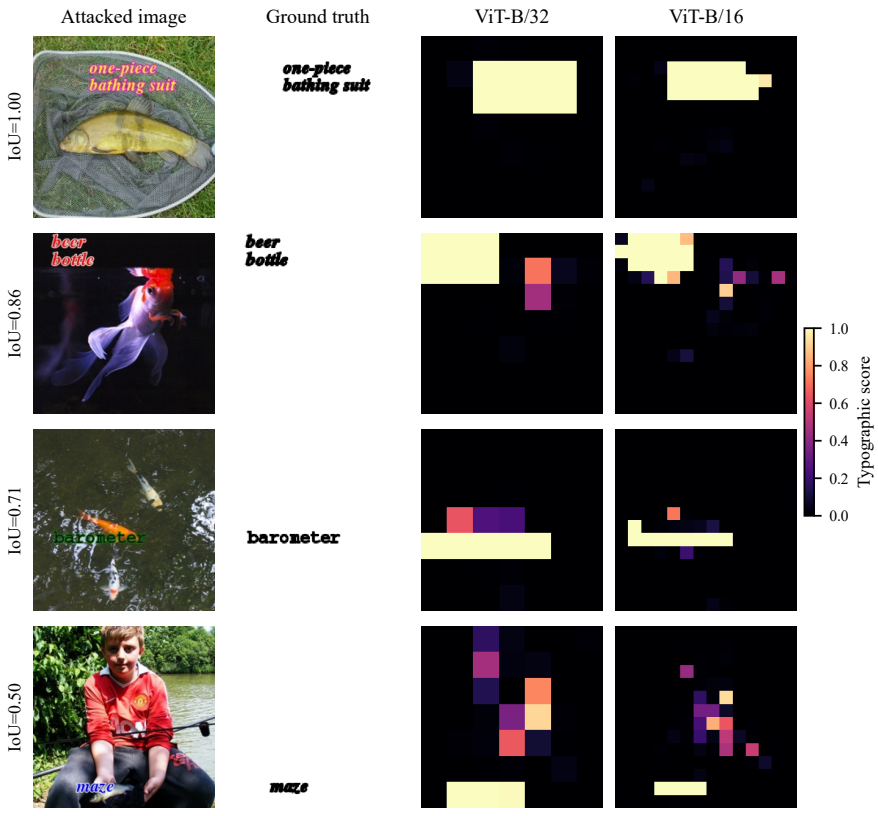


Figure 5: Typographic localization examples. Rows are selected automatically at the 0.90, 0.60, 0.30, and 0.10 quantiles of per-image ViT-B/32 IoU. Row labels report ViT-B/32 IoU at $p_i \geq 0.5$. Ground truth shows the reconstructed glyph mask; the last two columns show continuous prototype scores on the 7×7 and 14×14 patch grids.

E Additional Cross-Dataset Results

Tables 6 and 7 examine transfer from ImageNet-100 training to ten evaluation datasets covering objects, food, fine-grained categories, satellite images, textures, and scenes. The same ViT-B/32 projector and fixed prototypes are used across datasets. Table 6 compares clean and attacked recognition with frozen CLIP, and Table 7 compares attack performance with the other defenses.

Table 6: Zero-shot classification on ten clean datasets and their corresponding synthetic typographic-attack datasets using OpenAI CLIP ViT-B/32. Typographic entries report OCA/TCR (↑/↓). Gray values beside clean PTS accuracy report its absolute change from CLIP. For attacked PTS results, green and purple values report the absolute OCA gain and TCR reduction from CLIP, respectively; bold marks the better attacked OCA and TCR. Parenthetical changes are differences between the reported % values. The attacked ImageNet-100 configuration contains 101 candidate labels, whereas the clean split contains 100.

		Clean		Synthetic typographic attacks	
Dataset	N	CLIP	PTS	CLIP (OCA/TCR)	PTS (OCA/TCR)
ImageNet-100	5,000	82.68	82.24 (-0.44)	52.22/28.58	**73.62/1.32** (+21.40/-27.26)
Caltech-101	8,677	88.69	88.24 (-0.45)	64.09/26.83	**87.25/0.89** (+23.16/-25.94)
Food-101	25,250	84.14	84.02 (-0.12)	57.33/26.53	**79.84/0.24** (+22.51/-26.29)
Flowers-102	6,149	66.30	66.11 (-0.19)	30.80/39.65	**62.37/0.28** (+31.57/-39.37)
Oxford Pets	3,669	87.41	87.30 (-0.11)	59.96/20.11	**83.37/0.33** (+23.41/-19.78)
EuroSAT	8,100	43.10	43.04 (-0.06)	5.22/91.37	**53.65/5.59** (+48.43/-85.78)
DTD	1,880	44.57	44.57 (0.00)	26.01/38.14	**43.78/1.12** (+17.77/-37.02)
Stanford Cars	8,041	58.79	52.68 (-6.11)	20.37/24.30	**33.02/0.32** (+12.65/-23.98)
FGVC Aircraft	3,333	19.17	17.55 (-1.62)	10.65/15.75	**16.89/0.54** (+6.24/-15.21)
SUN397	19,850	61.71	61.08 (-0.63)	34.77/32.17	**58.92/0.05** (+24.15/-32.12)

Recovery across visual domains. Object recovery accompanies text suppression across all ten transfer sets. EuroSAT is the clearest stress case: frozen CLIP follows the rendered label on 91.37% of attacks, while PTS reduces TCR to 5.59% and raises OCA to 53.65% (Table 6). On Caltech-101, DTD, and SUN397, defended attacked accuracy approaches the corresponding clean accuracy, extending the pattern to objects, textures, and scenes. EuroSAT's attacked OCA even exceeds its clean PTS accuracy of 43.04% under the same class set and prompt. The scores depend on the input, so clean and attacked images receive different edits.

Clean recognition and remaining errors. Clean recognition stays close to CLIP on eight datasets, with larger losses on Stanford Cars and FGVC Aircraft (Table 6). On Cars, clean accuracy falls from 58.79% to 52.68%, yet the larger problem under attack is object recognition: TCR is only 0.32% while OCA is 33.02%. The remaining 66.66% of attacked images are classified as neither the correct car nor the rendered distractor. Most errors therefore lie among other object categories, separating successful suppression of the text from recovery of fine-grained visual distinctions.

Table 7: ViT-B/32 defenses on ten synthetic typographic-attack datasets. Each method occupies two rows for OCA (↑) and TCR (↓), both in %. Avg. is the unweighted macro-average over the ten datasets. Bold marks the best value for each metric and dataset. IN-100 denotes ImageNet-100. †: adapted discovery data, as in Table 3.

Method	Metric	IN-100	Caltech	Food	Flowers	Pets	EuroSAT	DTD	Cars	Aircraft	SUN	Avg.
CLIP	OCA	52.22	64.09	57.33	30.80	59.96	5.22	26.01	20.37	10.65	34.77	36.14
	TCR	28.58	26.83	26.53	39.65	20.11	91.37	38.14	24.30	15.75	32.17	34.34
D.-Prefix	OCA	67.20	79.84	68.97	44.48	74.54	9.98	30.69	28.06	14.52	43.91	46.22
	TCR	9.76	8.67	9.46	14.90	5.18	81.52	12.87	8.11	4.44	16.15	17.11
MirrorCLIP	OCA	69.22	84.15	76.37	50.43	77.24	25.51	35.59	33.04	14.22	53.36	51.91
	TCR	4.64	2.66	2.59	7.45	3.57	56.25	12.45	5.02	5.52	2.23	10.24
Dyslexify	OCA	68.98	82.69	73.09	52.33	78.96	19.35	37.82	**34.40**	16.50	50.07	51.42
	TCR	6.60	5.08	3.40	6.29	3.16	55.19	6.54	2.49	2.19	7.43	9.84
nTAS†	OCA	66.38	77.96	69.85	46.92	73.48	12.44	34.84	29.26	16.14	44.00	47.13
	TCR	10.00	10.40	9.21	12.07	5.31	75.95	10.90	4.04	2.79	16.17	15.68
PTS	OCA	**73.62**	**87.25**	**79.84**	**62.37**	**83.37**	**53.65**	**43.78**	33.02	**16.89**	**58.92**	**59.27**
	TCR	**1.32**	**0.89**	**0.24**	**0.28**	**0.33**	**5.59**	**1.12**	**0.32**	**0.54**	**0.05**	**1.07**

Comparison across defenses. PTS has the highest OCA on nine transfer datasets and the lowest TCR on all ten (Table 7). Its macro-average OCA reaches 59.27%, and TCR falls to 1.07%. Excluding EuroSAT preserves the OCA ranking over Dyslexify and MirrorCLIP, so the average advantage does not depend on the dataset most susceptible to rendered text.

On Stanford Cars, Dyslexify has higher OCA than PTS (34.40% vs. 33.02%), while PTS has lower TCR (0.32% vs. 2.49%). PTS suppresses the injected label more effectively but classifies fewer attacked cars correctly. Along with the clean-accuracy drop, this separates label suppression from fine-grained object recognition.